Application

For

United States Utility Patent

Title:

EXTREMELY SECURE METHOD FOR KEYING STORED CONTENTS TO A SPECIFIC STORAGE DEVICE

Inventor(s):

Christopher M. Carpenter, residing at 549 S. Frances Street, Sunnyvale, CA 94086, a citizen of the United States;

Todd Peter Carpenter, residing at 855 S. Lexington Parkway, St. Paul, MN 55116, a citizen of the United States;

John Masles, residing at 2753 Glauser Drive, San Jose, CA 95133, a citizen of the United States;

Chris Paul Dudte, residing at 5375 Saginaw Court, Reno, NV, 89433 a citizen of the United States.

EXTREMELY SECURE METHOD FOR KEYING STORED CONTENTS TO A SPECIFIC STORAGE DEVICE

Field of Invention

Invention relates to securing data in a storage medium device, more particularly to methods of securing specific files in a storage medium device to prevent use of unauthorized copies of the specific files.

Background of Invention

The relatively open and known architecture of a typical hard disk drive (HDD) renders it fairly easy for determined and minimally-funded attackers to duplicate content stored on the HDD. Low-level block copy software is easily available and produces an unauthorized drive image copy of the stored HDD content that is indistinguishable from the authorized source HDD for many host applications. Preventing a determined attacker from copying a drive's image to another drive and then using that copy on another host is difficult. Standard content encryption methods typically disallow viewing of the copied encrypted content, but it does not securely prevent the use of that content on another host having a valid decryption or usage key.

Typically, hardware authorization keys have been used to identify an authorized host. These keys have an added hardware cost and have historically been broken and duplicated in as little as a few days. This approach does not normally differentiate between source and copied contents. Other approaches to protect against unauthorized copying and / or use of disk contents typically require adding hardware to the host and / or disk drive to provide a secured or keyed communication channel and encrypted or keyed contents on the HDD. This approach generally adds hardware cost to the host and / or HDD. Also, this solution is not always transportable across HDD vendors because they

can require custom hardware. Moreover, copying encrypted contents to another HDD does not explicitly prevent its use. Another typical prior approach is requiring the original authorized CD-ROM to be physically present in a CD drive during use of the software or data. However, copying the original CD-ROM is easy. Thus, there is a need for a method to secure specific files to prevent the use of an unauthorized copied file stored on a storage medium.

Summary of Invention

An extremely secure method for keying source contents to a source storage medium is provided to prevent use of unauthorized copies, where there is no significant added cost to the disk drive. The host processor can use a well characterized encryption algorithm such as DES and a hard disk drive's (HDD's) statistically unique, immutable and verifiable physical attribute, such as the defect list, servo or channel characteristics to write a unique signature, or fingerprint, on a source medium. Accordingly, the extremely secure method of this invention allows use of source content with other similar hosts, but correspondingly disabling all copies of the sanctioned drive in any host.

The host processor reads the source medium original defect list or other such relatively immutable physical attribute. It then merges a representation of the attribute and the content to be secured. The host processor then encrypts the resulting content with a well-characterized algorithm such as DES. When a host wants to use the sanctioned source contents, it reads the source content from the storage medium and decrypts it with a decryption key. The host then parses the defect list out of the source content and explicitly reads the local storage medium defect list. If the resulting decrypted defect list matches the local storage medium defect list, then the host recognizes the local medium contents as sanctioned and the host continues use and processing of the source contents. If there is no match, then the local medium content is determined to be an unauthorized copy of the source storage medium. The host then rejects the use of the contents.

Brief Description of Drawings

Fig. 1 is a generalized block diagram of an extremely secure system for keying stored contents to a storage medium in accordance with the principles of the present invention;

Fig. 2A shows a generalized flowchart of an extremely secure method for keying stored contents to the storage medium in accordance with the present invention;

Fig. 2B shows a generalized flowchart of an extremely secure method for reading and verifying fingerprinted contents in a local storage medium in accordance with the present invention;

Fig. 3 shows a more expanded flowchart of the steps of storing fingerprinted contents of Fig. 2A;

Fig. 4 shows a more expanded flowchart of the steps of reading and verifying contents of Fig. 2B.

Detailed Description of Preferred Embodiment

During the manufacturing process, a hard disk drive (HDD) goes through a process of detecting media defects. These defects are represented by specific Physical Block Addresses (PBAs), collated and stored on the HDD in a structure called the "defect list," such as a "P-list," to insure that a host processor would never store user data in one of the defective PBAs. This list is immutable and does not change throughout the life of that drive. It is a physical, statistically unique, verifiable and relatively immutable (PSUVI) characteristic of that HDD. This list is an inherent physical signature that statistically differentiates each HDD from another.

Fig. 1 illustrated a generalized system block diagram 10 of an extremely secure system for keying stored contents to a storage medium in accordance with the principles

of the present invention. The extremely secure keying stored content system 10 comprises a host system 12 comprising a host interface 14 coupled to a host microprocessor 16, which is then coupled to other host system hardware generalized for simplicity here as general system 17. Host system 12 stores an extremely secure software application 100 to be later described in further detail with reference to Figs. 2-4. Extremely secure system 10 also comprises a disk drive unit 20 coupled to host unit 12 via a host-to-drive interface 18. Disk drive unit 20 comprises an interface and storage medium processor system 22, a servo system 24, a read/write system 26, one or more storage disks 30, and a preamplifier 28. Preamplifier 28 reads a PSUVI characteristic corresponding to, for example, "the defect list," or any other PSUVI associated with one or more storage disks 30. The read PSUVI characteristic is then used by host system 12 to encrypt a source content stored on the one or more storage disks 30.

Figs. 2A and 2B illustrate generalized flowcharts of extremely secured method 100 for keying stored contents to the storage medium (Fig. 2A) and for reading and verifying fingerprinted contents of stored information (Fig. 2B) in accordance with the present invention. In general as illustrated by this embodiment, in a first step 104 during a "storing fingerprinted contents" operation 102, a request is sent by host processor 16 to disk drive processor 22 to read a PSUVI characteristic, such as the defect list. During a second step 106, the read defect list is then combined with a specified file content to be secured to generate a fingerprinted content. In a step 146, the fingerprinted content can be encrypted first prior to storing. Then in step 108, host processor 16 then commands disk processor 22 to store the fingerprinted content on disk 30. During a "reading and verifying fingerprinted contents" operation 110, the host processor 16 commands the disk drive processor 22 to read fingerprinted content. In step 114, host processor 16 separates content and fingerprint. Subsequently, host processor 16 requests fingerprint from storage device 116. Then in step 118, host processor 16 compares content and storage device fingerprint. In a last step 120, the host processor 16 decides to use or not to use content based on comparison in step 118.

Fig. 3 illustrates in greater detail a sample method of storing fingerprinted content 102. In this example, host processor 16 would execute steps wherein host processor 16 requests a fingerprint from a storage device 20, such as a defect list from storage device 20, follow by step 106 wherein host 16 combines content of a file to be secured with the retrieved fingerprint, and step 108 wherein host 16 commands storage device to store fingerprinted content. As illustrated in more detail in Fig. 3, one embodiment to step 104 of requesting a fingerprint comprises:

1. Host 16 using open protocol to request secured communication from HDD in step 130;
2. HDD 20 identifies a PSUVI characteristic, such as a defect list in step 132;
3. HDD 20 then generates a decryption key and encryption key in step 134;
4. HDD 20 then returns encryption key to host 16 in step 136;
5. Host 16 then uses encryption key and switches to encrypted protocol in step 138;
6. Host 16 then requests fingerprint PSUVI characteristic 140; and then
7. HDD replies with PSUVI fingerprint in step 142.

As illustrated in more detail in Fig. 3, one embodiment to step 106 of combining content to be secured with the retrieved fingerprint comprises:

1. Host 16 creating a hybrid content by combining content and fingerprint 144; and
2. Host 16 encrypting hybrid content with public key 146.

Additionally, step 108 of storing fingerprinted content may comprise host 16 commanding HDD to write hybrid content 148.

Fig. 4 illustrates in greater detail a generalized method 110 of reading and authenticating a source content method of Fig. 2B. In this example, generalized method 110 of reading and verifying fingerprinted content comprises a first step 112 of a processor 16 commanding storage device 20 to read fingerprinted content. For

convenience of illustration, we assume processor used in this example is host 16. However, it is envisioned that the processor or host referred to and used herein to implement method 110 of reading and verifying source content can be generally a processor in any host system coupled to a storage device 20. Method 110 further comprises step 114 wherein host 16 separates file contents to retrieve the fingerprint content. Subsequently, in step 116, host 16 requests current storage device to provide fingerprint information. Host 16 then compares in step 118 fingerprint separated in step 114 with fingerprint retrieved in step 116 to verify fingerprints, and finally in step 120, host 16 then decides whether to use or not to use content based on the comparison step 118.

Fig. 4 further illustrates a sample detailed embodiment of steps described above for method 112 to read and verify fingerprinted contents.

1. Reading the defect list from the HDD (steps 160 and 162).
2. Decrypting the encrypted content. Parsing the vector subparts from the contents (steps 164-170)
3. Reassembling the subparts into a P-list vector (step 172).

More detailed implementation details for steps described in method 110 are provided also in Fig. 4 and are self-explanatory. Different possible embodiments of methods to verify authenticity of a copied file are envisioned and contemplated. The following described sample methods include using the defect list of a disk:

Signature Verification Method Example 1:

1. Perform low-level writes and reads on some or all of the PBAs in the defect list to determine whether or not read errors occurred at the supposed defect locations. Special microcode may be used to enhance the security of this verification step and protect from unauthorized interferences, or "man-in-the-middle" attacks. Well-characterized security methods for providing secured communications and

generating encryption keys from the HDD's unique signature, such as Diffie-Hellman, are well-documented in the security community and may be used herein.

2. Defects in the defect list do not necessarily have a probability of error equal to 1.0. Therefore the host would then determine that either a statistically large percentage of the P-list did point to defective PBAs and that the P-list was valid for the HDD, or that a statistically small percentage of the P-list pointed to defective PBAs and that the P-list was invalid for the HDD.

3. If the defect list was invalid, then the host would take steps to not use the HDD contents.

4. If the defect list was valid then the host would use the content.

Signature Verification Method Example 2:

If a unique signature other than the defect list is to be used, then the verification method changes accordingly. As an example, if Servo Burst Correction Values (BCVs) were used, measurement with BCVs turned on and off could indicate the validity of the HDD's current BCV values and whether or not they were altered. The same secured communications and key generation steps could be used to protect this verification algorithm.

Any items added to, or substituted for, the defect list in the algorithm prior to encryption fall into two categories:

PSUVI Characteristics: Relatively Immutable Physical Attributes Linkable to A Specific Head-Disk Assembly (HDA) or PCB. The signature attribute of this category is related to the statistically unique physical properties of the HDA or

electronics. A defect list falls into this category. These physical properties cannot be changed by a reasonable level of attack and can be measured by the drive. Servo wedge defects, BCV-related RRO responses, certain TMR behaviors, servo transfer functions and read or write channel optimization parameters related to individual heads also fall into this category. Any item in this category could substitute for the "defect list" above and satisfy the intent of this disclosure. The benefit of using a defect list based HDD differentiation is the low probability of any two HDDs having the same defect list and also that this list is physically verifiable, so that a change in the defect list is detectable.

Non-PSUVI Characteristics: Relatively Mutable Attributes Physically linked to a Specific Head-Disk Assembly (HDA) or PCB. Serial numbers on configuration pages, post-production defect list ("G-lists") and PROM contents fall into this category of non-PSUVI characteristics. These items are not statistically unique physical properties of the HDA or electronics, and they may be changed by an attacker with no secure method of verification. These attributes can be used, but typically require lengthening the encoded vector to statistically increase the time required for an attacker to break the encryption.

Key advantages of this invention are that no added hardware is necessary. This invention can be implemented using preexisting hardware, and can be implemented on existing hosts and HDDs. This invention deters against minimally to significantly funded unauthorized breaches or accesses of a secured content. Hosts, or local processors on hosts, can be responsible for security methods, rather than the drive. Moreover, this invention can be implemented with existing security methods.

The parts of this system that may require restricted access comprise the encryption / decryption keys and verification algorithms. Methods for encryption and access restriction are well documented in the security community. The specific algorithms for

encryption / decryption such as DES, or key generation algorithms such as Diffie-Hellman, are well-characterized and documented in the security community.

Sub A3

Foregoing described embodiments of the invention are provided as illustrations and descriptions. They are not intended to limit the invention to precise form described. In particular, it is contemplated that functional implementation of invention described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks. Other variations and embodiments are possible in light of above teachings, and it is thus intended that the scope of invention not be limited by this Detailed Description, but rather by Claims following.